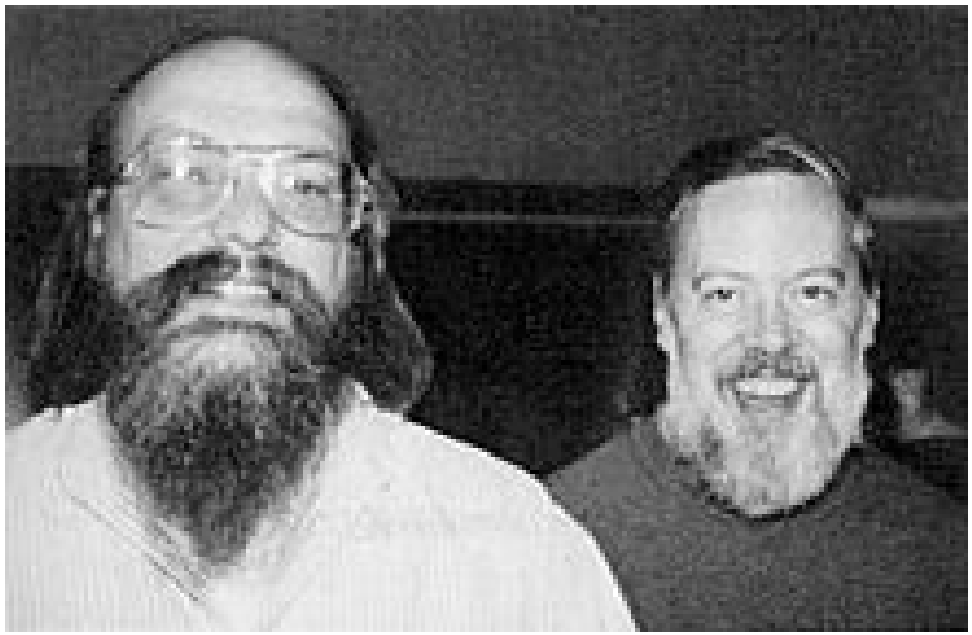
Introduction to Programming

Lecture One

N.R.Aravind

I.I.T. Hyderabad

12 Sep 2017



Ken Thompson and Dennis Ritchie

Dennis Ritchie: Creator of the C language

Wrote the Unix operating system

[Source:Internet]

Administrative

- Class webpage
  http://www.iith.ac.in/∼aravind/id1033
- Office hours: Tue, Wed AN (Off: 232, E-Block)
- Syllabus, grading policy, references, slides etc.
- Installation instructions

References

- The C Programming Language by Kernighan and Ritchie
- Practical C Programming by Steve Oualline

Grading Policy

- 15 marks: Attendance (Lab: 100%)
- Skip a lab $\Rightarrow$ Send me an email; attend another lab session the same week
- 20 marks: Assignments
- 30 marks: Final Exam (Written)
- 35 marks: Final Lab Exam
- No plagiarism

Lab Schedule

- Lab Attendance: 100%
- 210 and 219
- Wed: ME+ MA (**Tomorrow's lab rescheduled**)
- Thu: CS + ES
- Fri: CE + CH
- Mon: EP + MS

Lab Exercises

To be shared by email, coursepage

- Sample programs to read and execute
- List of programs for you to write

T.A.s (Teaching Assistants)

Topics in this lecture

- Introduction
- The Print Statement
- Variables and the Assignment statement
- Input statement

The C Programming Language

- Invented around 1970
- Operating systems like Linux
- Computer graphics
- For other languages (C is fast!)
- Embedded systems
- Solve engineering/scientific problems
- Games

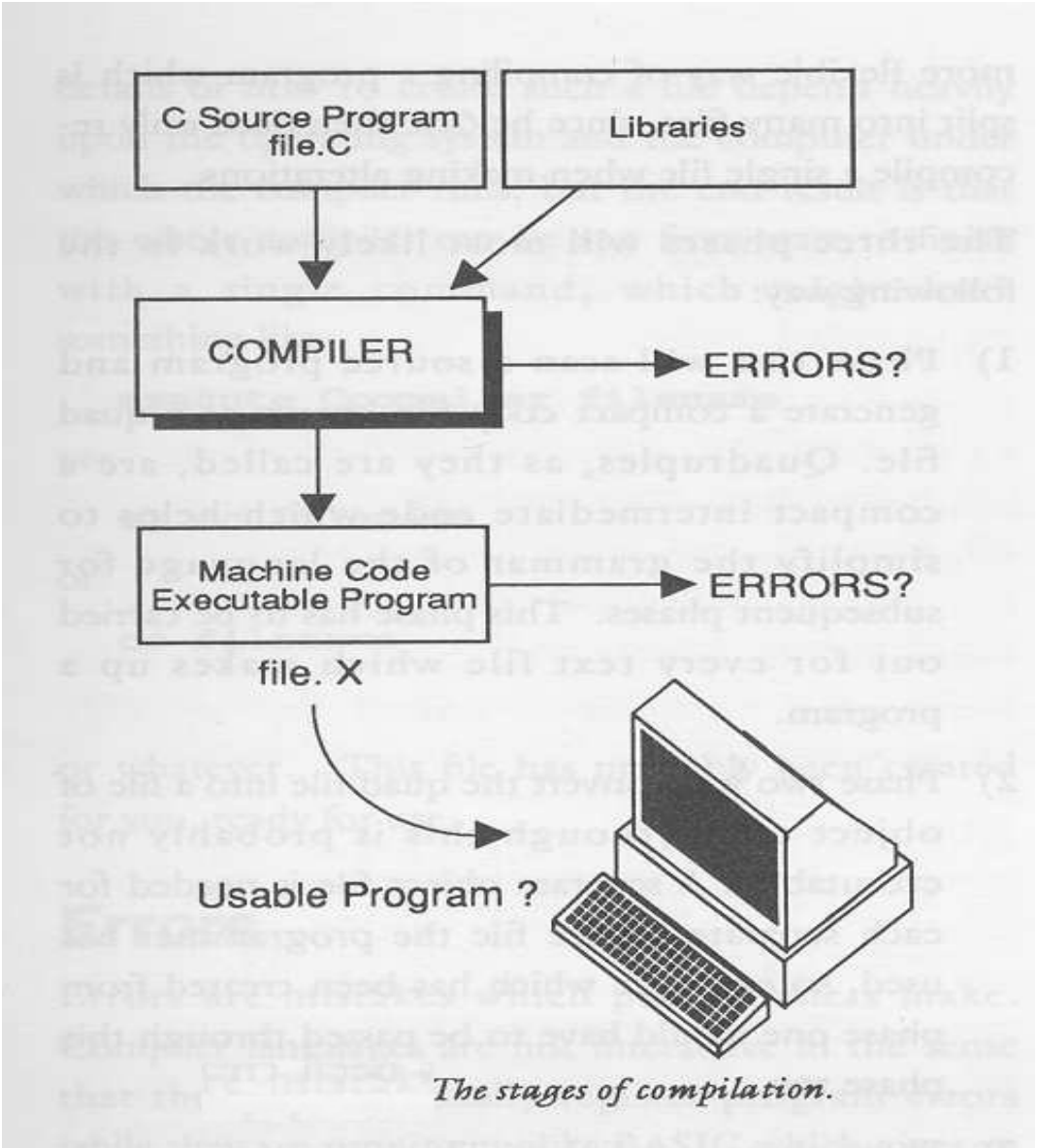
Machine Language

Source: Internet

Assembly Language

```
00000000                push    ebp
00000001                mov     ebp, esp
00000003                movzx   ecx, [ebp+arg_
00000007                pop     ebp
00000008                movzx   dx, cl
0000000C                lea     eax, [edx+edx]
0000000F                add     eax, edx
00000011                shl     eax, 2
00000014                add     eax, edx
00000016                shr     eax, 8
00000019                sub     cl, al
0000001B                shr     cl, 1
0000001D                add     al, cl
0000001F                shr     al, 5
00000022                movzx   eax, al
00000025                retn
```

Compilation

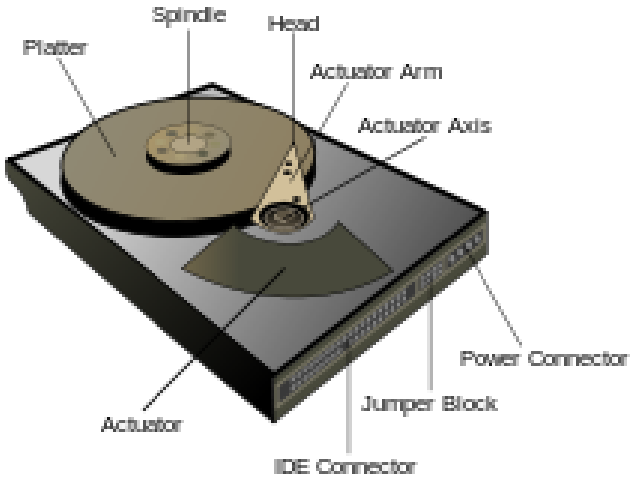


The stages of compilation.

```
000107: $R31.agc
000108: $P76.agc
000109: $R30.agc
000110: $STABLE_ORBIT.agc
000111: $BURN_BABY_BURN--MASTER_IGNITION_ROUTINE.ag
000112: $P40-P47.agc
000113: $THE_LUNAR_LANDING.agc
000114: $THROTTLE_CONTROL_ROUTINES.agc
```

Margaret Hamilton, Former Director of Software Engineering, MIT
Code for Apollo 11

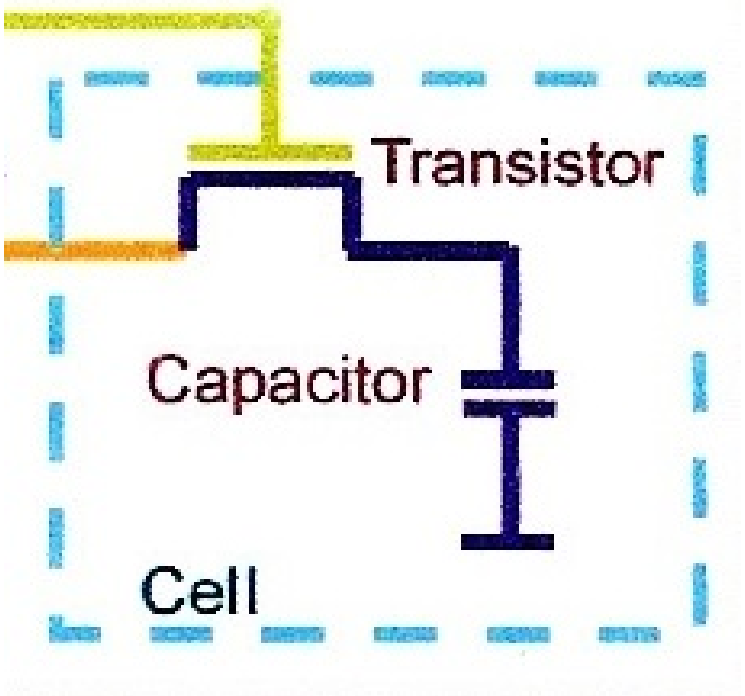
RAM (Random Access Memory) (Working memory)



Hard Disk (Secondary memory)

Source: Wikipedia

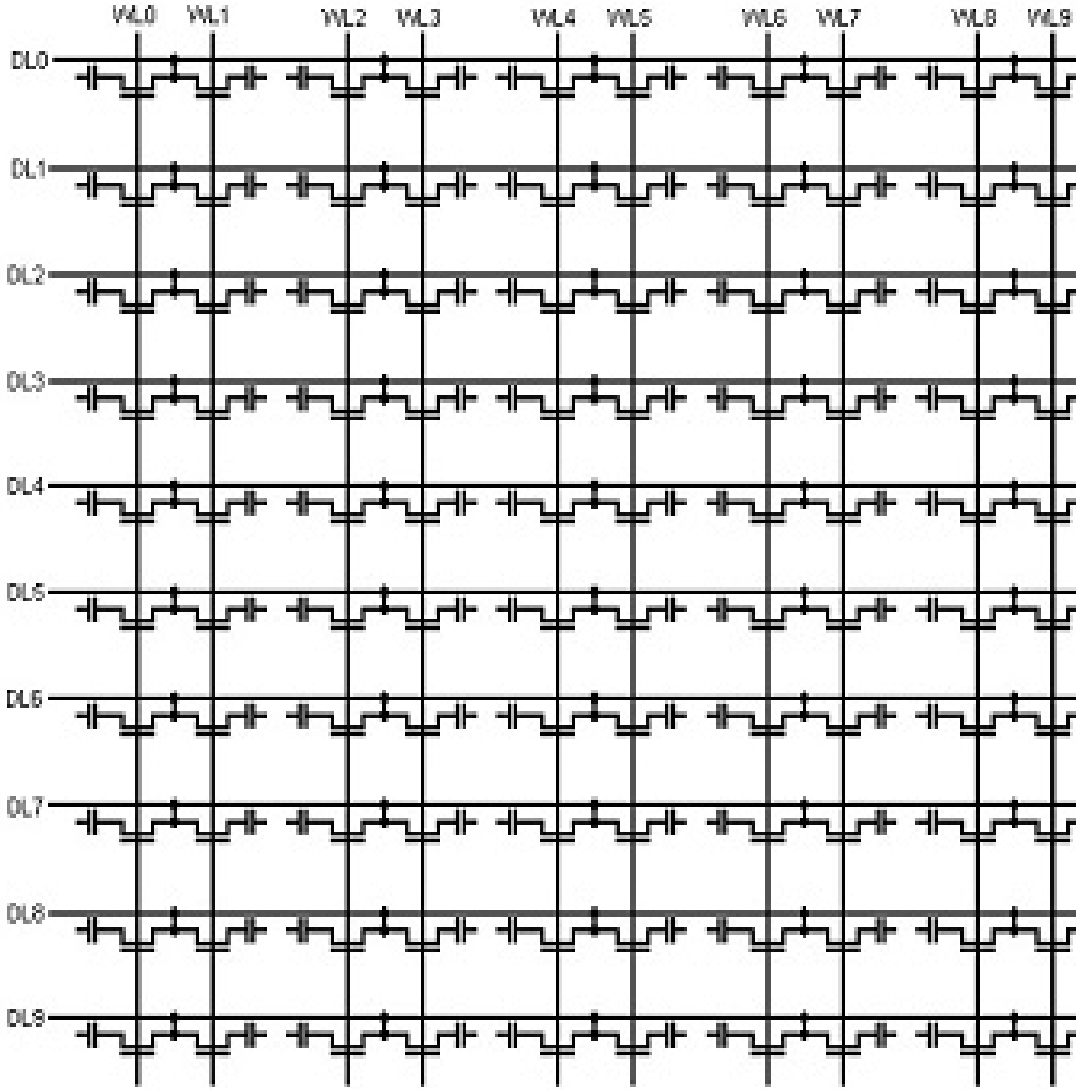
RAM Memory



Capacitor charge: 0 or 1
Transistor: Switch

source: internet

RAM Memory


WL0
WL1
WL2
WL3
WL4
WL5
WL6
WL7
WL8
WL9
DL0
DL1
DL2
DL3
DL4
DL5
DL6
DL7
DL8
DL9

Memory

0	0	0	1	0	0	0	1
1	1	1	1	1	1	1	1
0	0	1	0	0	0	1	1
0	0	0	0	0	1	1	0
0	0	0	1	1	1	0	1
1	1	0	0	1	0	1	0
			.	.	.		
			.	.	.		
			.	.	.		

Memory

0	0	0	1	0	0	0	1
1	1	1	1	1	1	1	1
0	0	1	0	0	0	1	1
0	0	0	0	0	1	1	0
0	0	0	1	1	1	0	1
1	1	0	0	1	0	1	0
			.	.	.		
			.	.	.		
			.	.	.		

Memory

0	0	0	1	0	0	0	1
1	1	1	1	1	1	1	1
0	0	1	0	0	0	1	1
0	0	0	0	0	1	1	0
0	0	0	1	1	1	0	1
1	1	0	0	1	0	1	0
			.	.	.		
			.	.	.		
			.	.	.		

Memory

2300	0	0	0	1	0	0	0	1
2301	1	1	1	1	1	1	1	1
2302	0	0	1	0	0	0	1	1
2303	0	0	0	0	0	1	1	0
2304	0	0	0	1	1	1	0	1
2305	1	1	0	0	0	0	1	0
2306				.	.	.		
2307				.	.	.		
2308				.	.	.		

A simple C program: helloWorld.c

```
#include<stdio.h>
main()
{
    printf("Hello, World");
}
```

A simple C program: helloWorld.c

```
#include<stdio.h>
// stdio.h: Standard input-output header file
// Contains declaration of printf
main() // Main point of execution
{
    printf("Hello, World");
     // printf("Some text");
}
```

Compiling and running a C program

```
$ gcc helloWorld.c -o hello
$ ./hello
Hello, World $
```

Compiling and running a C program

$ gcc helloWorld.c -o hello

- gcc: Gnu C Compiler
- Translates the C program into machine code named “hello”
- -o: specifies the outpiut file name

$ ./hello

- Run (execute) the program named "hello"
- To run a file named "xyz", type ./xyz.

Structure of C program

```
#include<  >
main()
{
Line 1
Line 2
Line 3
 .
 .
 .
}
```

Line 1 is first executed, followed by Line 2, followed by Line 3 etc.

Sequential execution

A simple C program: helloWorld.c

```
#include<stdio.h>
main()
{
    printf("Hello, World");
}
```

More about printf

```
#include<stdio.h>
main()
{
        char text[20]="Hello, World";
        printf("%s",text);
}
```

```
$./hello
Hello, World$
```

text:	H	e	l	l	o	,		W	o	r	l	d	\0

More about printf

```
#include<stdio.h>
main()
{
        char text[20]="World";
        printf("Hello, %s",text);
}
```

```
$./hello
Hello, World$
```

%s is replaced by the value in text.

More about printf

```
#include<stdio.h>
main()
{
    char movie[20]=“Interstellar”;
    char director[30]=“Christopher Nolan”;
    printf (“%s directed the film %s”, director,
movie);
}
```

What's the output?

$:Christopher Nolan directed the film Interstellar

The two %s are replaced by the values in director, movie resp.

The assignment statement

```
#include<stdio.h>
main()
{
      int a,b,c;
      a=10;
      b=20;
      c=a*b;
// a=10, b=20, c=400.
      printf("The value of a is %d",a);
      printf("\n The value of b is %d",b);
      printf("\n The value of c is %d",c);
```

The assignment statement

```
// a=10, b=20, c=400.
      a=a+b;
      b=a+b;
      c=c+1;
// a=? b=? c=?
// a=30 b=50 c=401
printf("\n The value of a is %d",a);
printf("\n The value of b is %d",b);
printf("\n The value of c is %d \n",c);
}
```

Variables in memory

0	0	0	1	0	0	0	1
1	1	1	1	1	1	1	1
0	0	1	0	0	0	1	1
0	0	0	0	0	1	1	0
0	0	0	1	1	1	0	1
1	1	0	0	1	0	1	0
			.	.	.		
			.	.	.		
			.	.	.		

Variables in memory

Address	Value
2300	17
2301	255
2302	35
2303	6
2304	29
2305	194
2306	.
2307	.
2308	.

int a,b,c;

Variables in Memory

	Address	Value
	2300	17
a	2301	255
	2302	35
	2303	6
b	2304	29
	2305	194
	2306	.
c	2307	.
	2308	.

```
int a,b,c;
```

Input: helloUser.c

```
#include<stdio.h>
main()
{
    char userName[20];
    printf(''Hello, user. What's your name? '');
    scanf("%s",userName);
}
```

Input a number

```
#include<stdio.h>
main()
{
    int num;
    printf(''‘Enter a number: ”);
    scanf(“%d”,&num);
}
```

TO BE CONTINUED...